REMARKS/ARGUMENTS

Claims 1-25 are pending in the present application. Claims 1-4, 7-8, 10-14, 17-18, and 20-25 were amended. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 101

The Examiner has rejected claims 11-20 under 35 U.S.C. § 101 as being directed towards non-statutory subject matter. This rejection is respectfully traversed.

The Examiner states:

> **Claim 11** recites a computer program product in a computer readable medium where support can be found in the specification on pg. 28. Specification discloses: the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media, such as a floppy disk, a hard disk drive, a RAM, CDROMs, DVD-ROMs, and transmission-type media, such as digital and analog communications links, wired or wireless communications links using transmission forms, such as, for example, radio frequency and light wave transmissions.
> Although, claim 11 is identified as a computer readable medium, the computer program product in a computer readable medium is directed to non-functional descriptive material that is capable of being distributed in the form of instructions and signals used to carry out distribution and transmission.

Office Action dated July 12, 2007, pages 2-3.

Applicant has amended claim 11 to recite: "A computer program product, which is stored in a computer recordable medium". In addition, Applicant has amended the specification to cancel the "transmission-type" media language. Therefore, the rejection of claims 11-20 under 35 U.S.C. § 101 has been overcome.

II. 35 U.S.C. § 103, Obviousness

The Examiner has rejected claims 1-25 under 35 U.S.C. § 103(a) as being unpatentable over *Pickett et al.*, Instruction Redefinition Using Model Specific Registers, U.S. Patent No. 6,076,156, dated June 13, 2000 (hereinafter referred to as "*Pickett*") and further in view of *Pechanek et al.*, Methods and Apparatus for Scalable Instruction Set Architecture with Dynamic Compact Instructions, U.S. Patent No. 6,848,041, dated January 25, 2005 (hereinafter referred to as "*Pechanek*"). This rejection is respectfully traversed.

The Examiner relies on *Pickett* to teach all features of Applicant's claims except the feature of dynamically setting a set of instructions using a selected instruction map. The Examiner asserts that *Pechanek* cures the deficiencies of *Pickett* and teaches dynamically setting a set of instructions using a selected instruction map. Applicant respectfully disagrees.

Pickett teaches redefining the operations performed by one or more instructions. One or more new instructions can be mapped to a re-definable instruction. A seldom-used instruction is one type of re-definable instruction. The microprocessor reassigns the operation performed by an instruction using a model specific register (MSR). The instruction redefinition mechanism becomes activated when an application program enables instruction redefinition.

The instructions that are re-definable can be preselected. Alternatively, the MSR may allow for storage of the opcodes corresponding to the instructions selected for redefinition, or a bit may be provided in the MSR that is assigned to each opcode. In this manner, the application itself may determine which instructions are re-definable.

Pechanek teaches a compacted instruction set, which allows a programmer the ability to dynamically create a set of compacted instructions. *Pechanek* teaches programmer loadable translation bits for translating instructions.

Applicant's independent claims recite a distinct set of patentable features. Claim 1 is representative of the other independent claims. Claim 1 recites: using, by an encryption algorithm each time the data processing system is rebooted, a different one of a plurality of different instruction maps to dynamically remap the standard instruction set to create a new instruction set; and processing, by the processing unit, only those instructions that use the new instruction set.

The combination of *Pickett* and *Pechanek* does not render Applicant's claims obvious because the combination does not teach or suggest using, by an encryption algorithm each time the data processing system is rebooted, a different one of a plurality of different instruction maps to dynamically remap the standard instruction set to create a new instruction set.

Neither *Pickett* nor *Pechanek* teaches either encryption or an encryption algorithm.

Pickett teaches preselecting instructions to be redefined. Since the instructions are preselected, they cannot be dynamically remapped. Therefore, this portion of *Pickett* teaches away from Applicant's claims.

Pickett also teaches an application determining which instructions are re-definable instructions. *Picket* does not teach the application being an encryption algorithm. Therefore, this portion of *Pickett* does not teach dynamically remapping, using an encryption algorithm.

Pechanek teaches a programmer that has the ability to dynamically create a set of compacted instructions. *Pechanek* does not teach the programmer being an encryption algorithm. Therefore, *Pechanek* does not teach dynamically remapping, using an encryption algorithm.

Therefore, the combination of *Pickett* and *Pechanek* does not teach using an encryption algorithm to dynamically remap an instruction set.

Applicant also claims using a different one of a plurality of different instruction maps to dynamically remap. The combination of *Pickett* and *Pechanek* does not teach or suggest this feature.

Pickett teaches redefining an instruction. *Pickett* also teaches several methods that can be used to achieve this redefinition. For example, an application can set bits within the MSR to redefine instructions. The redefinition is not achieved, however, using one of different instruction maps because *Pickett* does not teach a plurality of different instruction maps.

Pechanek does not cure this deficiency of *Pickett*. *Penchanek* teaches translation bits that are used to define a compacted instruction, but does not teach defining a compacted instruction using a different one of a plurality of different instruction maps because *Pechanek* also does not teach a plurality of different instruction maps.

Applicant also claims using, by an encryption algorithm each time the data processing system is rebooted, to dynamically remap. The combination does not teach or suggest this feature. Neither reference discusses rebooting the system or dynamically remapping each time the system is rebooted. Therefore, the combination of *Pickett* and *Pechanek* does not teach or suggest this feature.

Applicant's claims 2, 12, and 22 recite similar features. Claim 2 is representative of claims 12 and 22. Claim 2 recites: "performing the dynamic remapping during execution of an initial program load (IPL) process and before the data processing system begins executing an operating system". The combination of *Pickett* and *Pechanek* does not render these claims obvious because neither reference teaches an initial program load process.

In fact, *Pickett* teaches away from these features because *Pickett* teaches an application redefining an instruction. For an application to be able to redefine an instruction, the operating system must have begun executing. An application does not redefine during the initial program load process. Therefore, *Pickett* teaches away from performing the dynamic remapping during execution of an initial program load and before the data processing system begins executing an operating system.

Applicant's claims 4, 14, and 24 recite similar features. Claim 4 is representative of claims 14 and 24. Claim 4 recites: "encoding a set of instructions from a trusted computer base using the one of the plurality of different instruction maps to form a set of encoded instructions; and sending the set of encoded instructions to the processing unit for execution". The Examiner asserts that *Pickett*, column 4, lines 29-42, and column 5, lines 21-37, teaches these features. These sections are reproduced below.

> As opposed to preselecting the instructions to be redefined, microprocessor 10 may instead allow for storage of the opcodes corresponding to the instructions selected for redefinition. In this manner, the application itself may determine which instructions are re-definable instructions. Alternatively, a bit within the MSR may be assigned to each opcode. The application may thereby determine which instructions are remapped by setting the corresponding bit. Furthermore, preselected groups of instructions may be assigned to a particular bit.
>
> Decode unit 16 is coupled to receive signals from MSR unit 26 which represent the instruction redefinitions. For each instruction which microprocessor 10 allows to be redefined, decode unit 16 produces a first decoded instruction which causes execute units 18 and/or load/store unit 20 to perform the architecturally defined operation assigned to the instruction if the signals indicate that the instruction has not been redefined via an update to the corresponding MSR. Alternatively, decode unit 16 produces a second decoded instruction which causes execute units 18 and/or load/store unit 20 to perform a predefined operation which is different than the architecturally defined operation if the signals indicate that the instruction has been redefined via an update to the corresponding MSR.

Pickett, column 4, lines 29-46.

> Instruction cache 12 is a high speed cache memory for storing instructions. It is noted that instruction cache 12 may be configured into a set-associative or direct mapped configuration. Instruction cache 12 may additionally include a branch prediction mechanism for predicting branch instructions as either taken or not taken. Instructions are fetched from instruction cache 12 and conveyed to decode unit 16 for decode and dispatch to a reservation station 17.
>
> Decode unit 16 decodes each instruction fetched from instruction cache 12. Decode unit 16 dispatches the instruction to one or more of reservation stations 17 depending upon the type of instruction detected. More particularly, decode unit 16 produces a decoded instruction in response to each instruction fetched from instruction cache 12. The decoded instruction comprises control signals to be used by execute units 18 and/or load/store unit 20 to execute the instruction. For example, if a given instruction includes a memory operand, decode unit 16 may signal load/store unit 20 to perform a load/store memory operation in response to the given instruction.

Pickett, column 5, lines 21-41.

Nothing in the sections reproduced above teaches a trusted computer base. Therefore, the combination of *Pickett* and *Pechanek* does not render these claims obvious.

Applicant's claim 10 is representative of claim 20. Claim 10 recites: "wherein the new instruction set is created using a first one of the plurality of different instruction maps when code is executed by a first privilege level and wherein a second one of the plurality of different instruction maps is used when code is executed by a second privilege level". As discussed above, neither *Pickett* nor

Pechanek teaches a plurality of different instruction maps. Neither *Pickett* nor *Pechanek* teaches the other features of these claims. Therefore, the combination of *Pickett* and *Pechanek* does not render these claims obvious.

The remaining claims depend from one of the independent claims discussed above and are patentable for the reasons given above.

Therefore, the rejection of claims 1-25 under 35 U.S.C. § 103(a) has been overcome.

IV. Conclusion

It is respectfully urged that the subject application is patentable over the cited prior art and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: October 11, 2007

Respectfully submitted,

/Lisa L.B. Yociss/

Lisa L.B. Yociss
Reg. No. 36,975
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicant